



Zenoss Integration



- LightSpeed PM – A Certified Quest Partner



Updated 11-19-2019

Foglight Integration – Current OOTB Status

- Foglight does not come with an integration option for Zenoss out of the box, and requires custom services
 - Custom Event Driven Rule must be created
 - Command Invocation is required
 - Actions
 - Groovy Scripting
 - There is very Little control over event information
 - Control requires lots of groovy scripting
 - There are no options offered or available to define Zenoss Fields in Foglight

Zenoss Custom Integration Cartridge – Highlights

- Our Zenoss Integration Cartridge simplifies integration with Foglight:
 - Advanced Rich UI for all functionalities
 - Customizable Command Line invocation support
 - Customizable REST invocation support
 - Create/Update Tickets
 - Flexible parameter definitions
 - Auto Annotate Alarms with returned information
 - **Remote Administration** of solution across servers
 - Decision engine for total and precise control
 - 20 Levels of granularity
 - Severity Level control

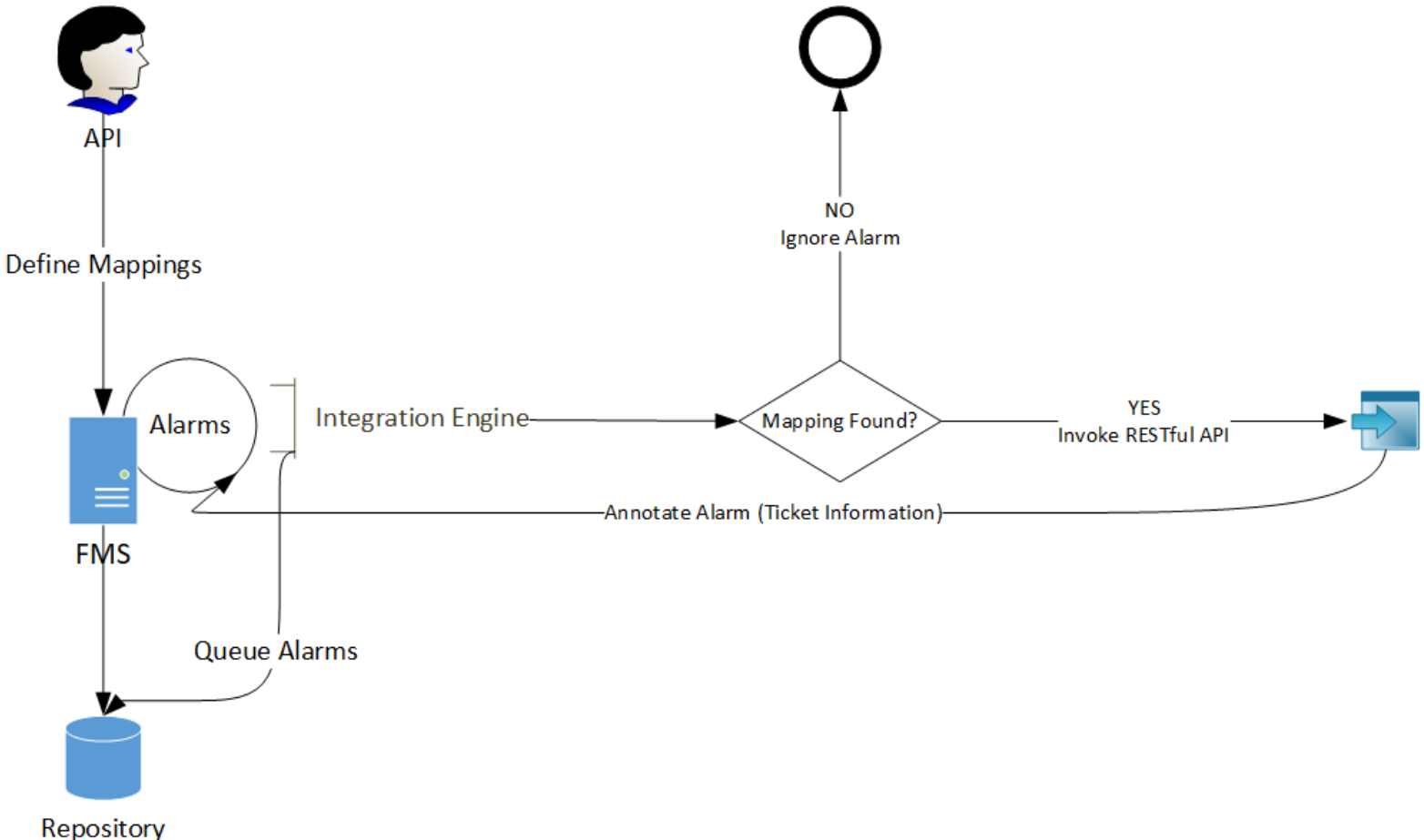
Zenoss Integration – Highlights

- Pattern editor to easily access many of the information related to the alarm. (No groovy required)
 - Access to dozens of fields to easily provide values to parameters
 - A dozen common operators available to easily avoid any coding
- Use of groovy scripts for power users
 - Use of groovy scripting with complete injected flow context
 - Ability to reject/delay ticket generation
- Persisted Queue so no events are lost
 - Guaranteed delivery
 - Keep history for rejected, failed and successful operations for total control
 - Pause or delay control
 - Consumer Thread settings/control as to manage concurrency and volume.

Zenoss Integration – Highlights

- Alarm Transition Support
 - Can be turned ON or OFF
 - Update tickets when alarm is transitioning from one severity to another.
 - Can be either to higher severity only or any transition
 - Close ticket only when the last alarm in the transition has been cleared
 - All transition information is accessible from the Pattern Editor
- Alarm Flood Prevention
- Auto Acknowledge Alarms

Zenoss Integration – RESTful API Flow



Zenoss Integration – Decision Engine

- The decision engine is based on various mappings defined within the solution. A very rich UI is provided to create the mappings.

• Low Priority

	Service	Host	Agent Type	Agent	Topology Object	Rule
0						
1		x				
2			x			
3				x		
4						x
5					x	
6		x				x
7			x			x
8				x		x
9					x	x
10	x					
11	x	x				
12	x		x			
13	x			x		
14	x					x
15	x				x	
16	x	x				x
17	x		x			x
18	x			x		x
19	x				x	x

No Service

With Service

• High Priority

Zenoss Integration – Changing Settings

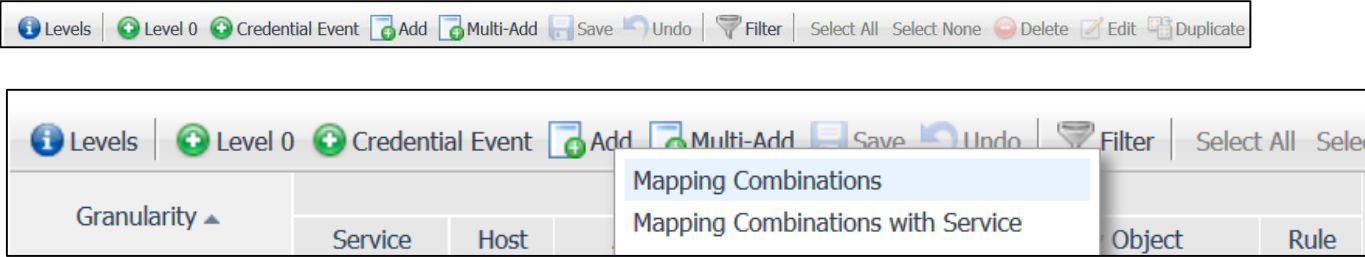
Integration Pack Registry Settings - Local FMS

Save | Undo | Filter

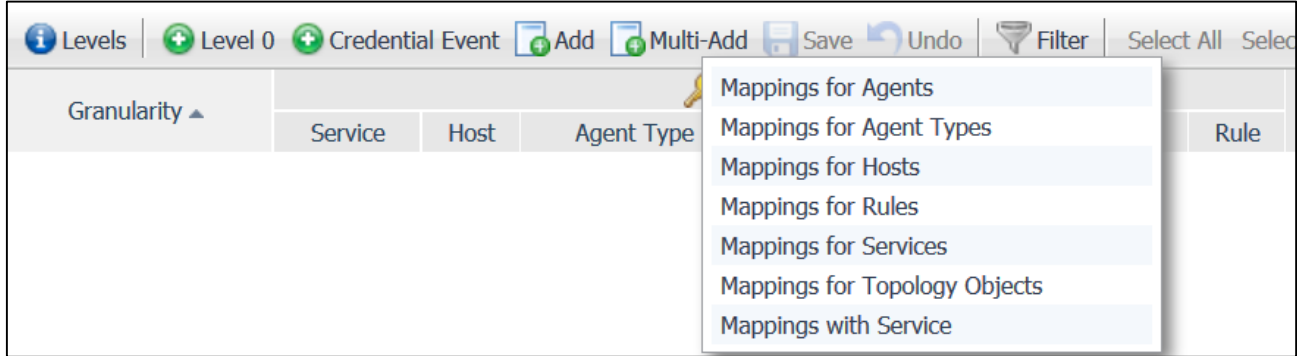
Name	Value	Scoping	
Zenoss Integration			
Open Ticket			
PSO.ZenossIntegration.OpenTicket.NoEmptyValue	true		If true any tag that returns a null will be replaced by a space. If false null values are simply not outputted.
PSO.ZenossIntegration.OpenTicket.TimeZone	GMT		Defines the time zone to use when outputting date and time in the Command.
PSO.ZenossIntegration.OpenTicket.DateTimeFormat	dd/MM/yyyy:HH:mm:ss		Defines the format for the date and time to use when outputting date and time in the Command.
Queue			
Ignore			
PSO.ZenossIntegration.Queue.Ignore.Normal	true		Should cleared alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall queue size.
PSO.ZenossIntegration.Queue.Ignore.Warning	false		Should Warning alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall queue size.
PSO.ZenossIntegration.Queue.Ignore.Critical	false		Should Critical alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall queue size.
PSO.ZenossIntegration.Queue.Ignore.Fatal	false		Should Fatal alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overall queue size.
PSO.ZenossIntegration.Queue.Ignore.Rules			A list of comma separated rule names that the queue should ignore. Alarms generated by ignored rules are not queued.
Delay			
PSO.ZenossIntegration.Queue.Alarm.DelayPeriod.Warning	0		Time in minutes a warning alarm should be delayed in the queue before being forwarded to the target system if the target system is unavailable.
PSO.ZenossIntegration.Queue.Alarm.DelayPeriod.Critical	0		Time in minutes a critical alarm should be delayed in the queue before being forwarded to the target system if the target system is unavailable.
PSO.ZenossIntegration.Queue.Alarm.DelayPeriod.Fatal	0		Time in minutes a fatal alarm should be delayed in the queue before being forwarded to the target system if the target system is unavailable.
PSO.ZenossIntegration.Queue.AlarmTransition.Enable	false		Should alarm transition be supported. If enabled the integration will update a ticket vs creating a new one when an alarm transitions.
PSO.ZenossIntegration.Queue.Alarm.AcknowledgeUponSuccess	false		Should the alarm be acknowledged when the alarm has been successfully processed.
PSO.ZenossIntegration.Queue.Alarm.EnforceServiceFilter	false		When an alarm is being processed enforce the filters applied to the service. If true a service is not considered to be in a failed state until it is explicitly acknowledged.
PSO.ZenossIntegration.Queue.PausePeriod	0		Time in minutes no alarms should be forwarded to the target system. This is useful when the target system is unavailable.
REST			
PSO.ZenossIntegration.REST.Enable	false		Should the integration use the REST method (true)
PSO.ZenossIntegration.REST.URL			The URL for the RESTful API the integration should use in order to open a ticket.
PSO.ZenossIntegration.REST.Path			The path following the URL.
PSO.ZenossIntegration.REST.Update.Path			The path following the URL to use when an alarm is cleared or if Alarm Transition is enabled when an alarm transitions.
PSO.ZenossIntegration.REST.Update.Method	POST		Which method to use when doing an update: POST or PUT
PSO.ZenossIntegration.REST.ResponseProperty			The name of the property to retrieve the Zenoss ticket ID after opening a ticket.
PSO.ZenossIntegration.REST.OtherResponseProperties			A comma separated list of properties to retrieve from the REST response. These properties are secondary to the ticket ID.
PSO.ZenossIntegration.REST.TrustAllSSLCerts	false		If true all SSL certificates will be trusted. Should only be used in Development environments where certificates are not trusted.
Authentication			
PSO.ZenossIntegration.REST.AuthenticationMode	HTTP_HEADER		HTTP_HEADER, BASE64_ENCODING or URL_ENCODE authentication when not tokens are needed, or BASE64_ENCODING when tokens are needed.
PSO.ZenossIntegration.REST.Username			The username to gain access to the RESTful API the integration should use in order to open a ticket.
PSO.ZenossIntegration.REST.Password	*****		The password to gain access to the RESTful API the integration should use in order to open a ticket.
PSO.ZenossIntegration.REST.Authentication.URLEncodeFormat			The format in which to send the login information. property={username} or {password}. It is possible to add other properties to the login information.
PSO.ZenossIntegration.REST.Authentication.TokenURL			The URL to use for authentication. Should be used if a 3rd party login needs to be used. Use only if AuthenticationMode is set to TOKEN.

Zenoss Integration – Mappings Editor

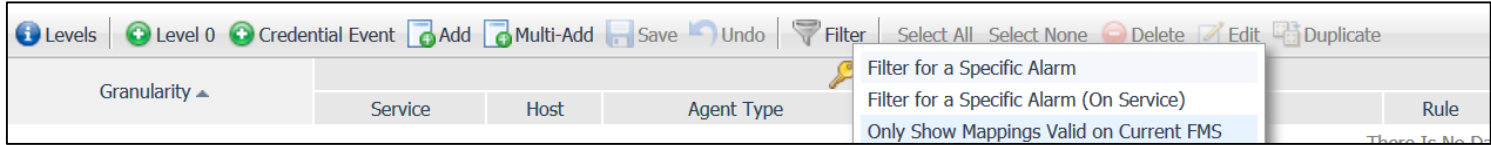
- Combinations



- Multiple Mappings



- Filtering



- . . .

Zenoss Integration – Field Editor

Corrective Message :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Duration :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Expires On :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Message :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Severity :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Source :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Timestamp :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Type :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Uuid :	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="button" value="↶"/>
Enable Normal <input checked="" type="checkbox"/> : true	<input type="button" value="⏻"/>	<input type="radio"/>	<input type="button" value="↶"/>
Warning <input checked="" type="checkbox"/> : true	<input type="button" value="⏻"/>	<input type="radio"/>	<input type="button" value="↶"/>
Critical <input checked="" type="checkbox"/> : true	<input type="button" value="⏻"/>	<input type="radio"/>	<input type="button" value="↶"/>
Fatal <input checked="" type="checkbox"/> : true	<input type="button" value="⏻"/>	<input type="radio"/>	<input type="button" value="↶"/>

Zenoss Integration – Pattern Editor

Alarm Severity Name Elements

Add Insert Edit Complete Save Undo

Pattern Element

- @ALARM_SEVERITY_NAME

Remember to click Apply if you manually edit the pattern.
Spaces are automatically trimmed, please use Edit Complete to insert spaces and have them preserved.

@ALARM_SEVERITY_NAME

Apply

TAGS

Tag	Caption
AGENT_TYPE	Agent Type
AGENT_UID	Agent UID
ALARM_ACKNOWLEDGED_BY	Alarm Acknowledged By
ALARM_ACKNOWLEDGED_TIME	Alarm Acknowledged Time
ALARM_CLEARED_BY	Alarm Cleared By
ALARM_CLEARED_TIME	Alarm Cleared Time
ALARM_CREATED_TIME	Alarm Created Time
ALARM_ID	Alarm ID
ALARM_IS_ACKNOWLEDGED	Is Alarm Acknowledged
ALARM_IS_CLEARED	Is Alarm Cleared
ALARM_LINK	Alarm Link

OPERATORS

Operator	Short Description
?[]	If tag has a value.
[]	If tag does not have a value.
?[]:[]	If tag has a value else.
^{}{}	Replace string with another.
=0	Equality
=0?[]:[]	If equal else
![[]]	Groovy Script
^U	Uppercase
^L	Lowercase
^T	Trim
^N	New Line Output
^R	Return

Zenoss Integration – System Requirements

- **Minimum required FMS version**

5.9.3

- **Supported Databases**

Microsoft SQL

Oracle

MySQL

PostgreSQL

Minimum Version

2008 (version 10.0.1600 or later)

9i R2

5.1.45

9.4.0